Шановний голово, шановні члени державної екзаменаційної комісії, шановні присутні. Вашій увазі пропонується дипломна робота освітньо-кваліфікаційного рівня «Магістр» на тему «Табу-пошук для квадратичної задачі про призначення».

Квадратична задача про призначення була вперше сформульована у 1957 році Купменсом та Бекманом. Задача моделює наступну проблему із реального життя: дано множину із *N* об'єктів та множину із *N* місцеположень. Для кожної пари місцеположень задана відстань і для кожної пари об’єктів задана вага або потік (тобто кількість поставок, що транспортуються між двома об’єктами). Задача полягає в розташуванні всіх об’єктів у різних місцеположеннях з метою мінімізації суми відстаней, помножених на відповідні потоки.

Формальна постановка задачі наступна: дано дві множини – *F* (об’єкти) та *L* (місцеположення) рівних потужностей, на яких визначена функція ваги *w : F × F → R* та функція відстані *d : L × L → R*. Необхідно знайти таке відображення *f : F → L* (призначення), що мінімізує цільову функцію

Задача є NP-складною, тому не існує алгоритму її розв’язання за поліноміальний час. Задача комівояжера може розглядатися як частковий випадок квадратичної задачі про призначення, якщо потоки з’єднують об’єкти лише вздовж кола і мають однакове значення (константу).

До практичних застосувань квадратичної задачі про призначення включають:

1. задача розташування заводів та фабрик;
2. задача розміщення електронних компонентів на печатній платі або мікросхемі;
3. задача проводки електроживлення;
4. задача розташування кампусів або корпусів лікарень;
5. задача проектування розкладки клавіатури та ін.

У даній роботі представлено опис знаходження нової властивості квадратичної задачі про призначення, а також розроблене на її основі математичне підґрунтя та табу-пошук.

Отже, розв’язком задачі є вектор, елементами якого є номери об’єктів. Розв’язком, з якого починається пошук, є вектор із довільною перестановкою чисел. Перед початком пошуку один раз зі складністю обчислюється вартість даного випадкового розв’язку за формулою

Далі табу-пошук працює з околом сусідніх розв’язків, які отримуються шляхом перестановки місцями двох елементів у векторі. Відомо, що вартість сусіднього розв’язку дорівнює вартості попереднього розв’язку + деяке число ∆, яке обчислюється за формулою

де *i* та *j* – це індекси елементів у векторі , що переставляються місцями.

Очевидно, що складність такої операції становить у зв’язку зі знаком суми. Перед початком пошуку заздалегідь обчислюються всі значення для перестановок у векторі всіх можливих пар елементів з індексами *i* та *j*, де . Всього таких перестановок , тобто . На кожній ітерації табу-пошук обирає **незаборонену** перестановку із мінімальним значенням ∆, тобто виконується перевірка перестановок. Обрана перестановка двох елементів здійснюється і до значення вартості попереднього розв’язку додається ∆. При цьому оновлюються всі обчислені раніше значення .

1. Очевидно, що нове значення для тих самих елементів *i* та *j*, що щойно були переставлені, тепер буде дорівнювати .
2. Нові значення , на які змінюється вартість розв’язку внаслідок перестановки елементів з індексами *p* та *q*, відмінними від *i* та *j*, обчислюються зі складністю за відомою формулою
3. Проте значення для перестановок елементів з індексами *r* та *s*, один з яких дорівнює індексу *i* або *j*, обчислюється повністю заново зі складністю за вже знайомою нам формулою

де замість *i* та *j* тепер записано *r* та *s* відповідно.

Всього кількість значень , які заново обчислюються зі складністю , становить . Це видно зі **слайду**, на якому проілюстровано можливу перестановку кожного іншого елемента *k* з одним елементом із пари щойно переставлених місцями елементів. Тоді кількість перестановок, що оновлюються зі складністю дорівнює різниці кількості всіх можливих перестановок та тих, що обчислюються зі складністю :

У даній роботі вперше було запропоновано **техніку** оновлення половини значень з обчислювальною складністю замість . Дана техніка ґрунтується на новій властивості квадратичної задачі про призначення, що була **знайдена** в даній роботі.

Розглянемо **ускладнений випадок**, коли щойно у векторі були переставлені місцями елементи та і **необхідно** обчислити нові значення та . Розглянемо добутки відстаней на відповідні потоки від місцеположень *i*, *j*, *k* до кожного місцеположення із решти місцеположень.

1. Розглянемо спочатку відповідність відстаней потокам до здійснення будь-яких перестановок (**слайд**). При цьому припускаємо, що відомі значення , , . До даних відомих входять такі складові, що стосуються решти незмінних призначень (**слайд**).
2. Тепер розглянемо відповідність відстаней потокам після здійснення перестановки елементів  та  (**слайд**). Перед нами постає задача обчислити нові значення до яких входять такі складові, що стосуються решти статичних призначень (**слайд**).

Слід відзначити, що саме ці складові зумовлюють складність обчислення нових значень . Після проведення дослідження правих частин складених виразів була здійснена спроба пошуку залежностей між ними, в результаті чого було **знайдено** взаємозв’язок, що описується формулою

Цінність знаходження даної залежності полягає в тому, що, використавши її, можна обчислювати лише одне значення або зі складністю , а друге – зі складністю .

Однак, до , , , ,  окрім досліджених вище складових входять також інші складові , кожна з яких враховує нерозглянуті відстані та потоки між самими місцеположеннями *i*, *j*, *k* та об’єктами , , :

Отже можемо подати формулу для взаємозв’язку шуканих та у вигляді

Тоді формула для обчислення нового значення зі складністю через вже обчислене нове значення матиме вигляд

Якщо підставити вирази для кожного значення *R*, то часткова сума значення

матиме такий дещо громіздкий вигляд (**слайд**). Для отримання альтернативного представлення даної формули було написано MATLAB скрипт, в результаті запуску якого формула спростилася до вигляду

Далі з’ясуємо, на скільки суттєвим є **знаходження** даної властивості квадратичної задачі про призначення. Для цього засобами Visual Studio проведено Profiling табу-пошуку із застосуванням виведеної формули складності , та без її використання, коли всі ∆ обчислюються зі складністю .

Як видно зі **слайду**, більше 50% всього часу пошуку раніше йшло на повне обчислення нових значень ∆. Тепер на обчислення лише половини значень ∆ зі складністю , а решти – зі складністю , йде 36% загального часу.

Тепер перейдемо до чисельних результатів роботи нового табу-пошуку на основі розробленого математичного **підґрунтя**.

У таблиці наведений середній час пошуку глобального оптимуму для алгоритмів розробленого табу-пошуку, імітації відпалу, Fast Ant System, Robust Tabu Search та Алгоритму перетину локальних оптимумів, який був темою моєї доповіді на ПМК-2011 і зазнав модифікації із застосуванням розробленого математичного **підґрунтя**. Як бачимо, табу-пошук та алгоритм перетину локальних оптимумів поступилися за часом пошуку лише алгоритму Fast Ant System для двох із 9 задач, проте йому взагалі не вдалося знайти глобальний оптимум для задачі *Tai25a*.

Далі наведено таблицю із середнім значенням вартості розв’язку за 100 мс пошуку. У даному випадку спостерігається схожа картина: алгоритм Fast Ant System показав кращий результат лише для задачі *Tai30b*. Для всіх інших задач найефективнішими виявились розроблені алгоритми.

Слід також відзначити, що абсолютно у всіх випадках розроблений табу-пошук показав себе більш ефективним, ніж алгоритм Robust Tabu Search, який є одним із найбільш ефективних у своєму класі.

Отже, наукова новизна результатів проведених досліджень наступна:

1. знайдено нову властивість квадратичної задачі про призначення, яка відображає взаємозв’язок всіх значень ∆, на які змінюється вартість розв’язку внаслідок потенційних перестановок пар елементів.
2. вперше виведено формулу, яка дозволяє зі складністю обчислювати половину значень, які раніше обчислювались зі складністю .
3. розроблено новий табу-пошук, який характеризується до двох разів вищою швидкодією та якістю розв’язків, ніж інші евристичні алгоритми цього класу.

На цьому доповідь завершено, дякую за увагу.